Conditional Statements in Python

Decision Making (If statement)

- Decision structures evaluate multiple expressions which produce TRUE or FALSE as outcome.
- Python assumes any **non-zero** and **non-null** values as TRUE, and if it is either **zero** or **null**, then it is assumed as FALSE value.

```
if x < y:
    print('less than')

if y:
    print('non zero value')

if x or y:
    print('or operator')

if 'cc' in 'rcciit':
    print('substing')

if 'mnp' in ['abc', 'mnp', 'xyz']:
    print('in operator')
```

In C, the curly braces { } are used for more than one statement under IF statement. But in Python, we should use **indentation**.

```
if 20 < 10:
    print('Welcome')
print('Bye')
Output:  Bye
if 10 < 20:
    print('Welcome')
    print('Bye')
Output: Welcome
        Bye
```

Decision Making (else and elif Clauses)

```
if x > y:
    print('x is maximum')
else:
    print('y is maximum')
```

```
if x >= 60:
    print('First Class')
elif x >= 40:
    print('Second Class')
else:
    print('Fail')
```

```
if x > y:
    if x > z:
        print('x is maximum')
    else:
        print('z is maximum')
else:
    if y > z:
        print('y is maximum')
    else:
        print('z is maximum')
```

Loop Statements(While Loop)

```
count = 1
while (count <= 5):
    print(count)
    count = count + 1
Output: 1
        2
        3
        4
        5
```

- Python supports to have an else statement associated with while loop.
- Else statement is executed when the condition becomes false.

```
count = 1
while (count <= 5):
    print(count)
    count = count + 1
else:
    print('Bye')
Output: 1
        2
        3
        4
        5
        Bye
```

Loop Statements(For Loop)

```
for letter in 'COVID':
        print(letter)
```

Output: C O V I D

(each letter in different line)

```
months=['Jan', 'Feb', 'Mar']
for month in months:
        print(month)
```

Output: Jan Feb Mar

(each month in different line)

```
for count in range(5):
       print(count)
```

Output: 0 1 2 3 4

(each number in different line)

```
for count in range(1,7,2):
       print(count)
```

Output: 1

3

5

```
months=['Jan','Feb', 'Mar']
for index in range(len(months)):
       print(months[index])
```

Output: Jan

Feb

Mar

The break Statement

- The **break** statement can be used in both *while* and *for* loops.
- It terminates the current loop and resumes execution at the next statement

```
for count in range(1,10):
    if count==5:
        break
    print(count)
```

Output: 1 2 3 4 (each number in different line)

```
for i in range(1,3):
    for j in range(5,10):
        if j==8:
            break
        print(j)
```

Output: 5 6 7 5 6 7 (each number in different line)

The continue Statement

- The **continue** statement rejects all the remaining statements in the current iteration of the loop and moves the control back to the top of the loop.

```
for count in range(1,10):
    if count==5:
        continue
    print(count)
```

Output: 1 2 3 4 6 7 8 9 (each number in different line)

```
for i in range(1,3):
    for j in range(5,10):
        if j==8:
            continue
        print(j)
```

Output: 5 6 7 9 5 6 7 9 (each number in different line)

The pass Statement

- The **pass** statement is a *null* operation; nothing happens when it executes.

```
for count in range(1,6):
        if count==3:
                pass
                print('This is pass statement')
        print(count)

Output:  1
         2
         This is pass statement
         3
         4
         5
```

THANK YOU